

PCT/JP 03/16231

18.12.03

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 2 年 1 2 月 2 0 日

出 願 番 号
Application Number: 特 願 2 0 0 2 - 3 7 0 9 5 3
[ST. 10/C]: [J P 2 0 0 2 - 3 7 0 9 5 3]

出 願 人
Applicant(s): 澤 田 研 一

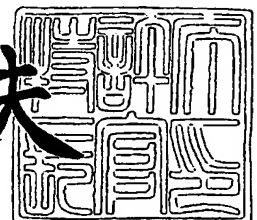
PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

RECEIVED
12 FEB 2004
WIPO PCT

2 0 0 4 年 1 月 2 9 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 4 - 3 0 0 3 9 0 3

【書類名】 特許願
【整理番号】 15951
【提出日】 平成14年12月20日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 12/00
G06F 13/00
G06F 15/00

【発明者】
【住所又は居所】 東京都港区白金 1 - 2 5 - 2 4 - 3 0 5

【氏名】 澤田 研一

【特許出願人】
【識別番号】 502248555
【氏名又は名称】 澤田 研一

【代理人】
【識別番号】 100082670
【弁理士】
【氏名又は名称】 西脇 民雄

【選任した代理人】
【識別番号】 100114454
【弁理士】
【氏名又は名称】 西村 公芳

【手数料の表示】
【予納台帳番号】 007995
【納付金額】 21,000円

【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 ウェブアプリケーションサーバ用支援プログラム及びサーバ

【特許請求の範囲】

【請求項 1】

通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される一連の処理の処理ステップに従って、前記返信情報に基づく処理を実行するウェブアプリケーションサーバの制御部に、

前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、

前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信させる情報送信処理と、

前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、

受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出させる返信情報検出処理と、

前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行させ、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライア

ト特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行させる業務実行処理と
を実行させることを特徴とするウェブアプリケーションサーバ用支援プログラム。

【請求項 2】

前記記憶手段は、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目毎のデータを記憶するデータ記憶部とを備え、

前記制御部に対して

前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合には、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶させ、

前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項 1 に記載のウェブアプリケーションサーバ用支援プログラム。

【請求項 3】

クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信手段を用いてクライアントに送信し、該通信手段において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、前記一連の処理において使用される設定項目別のメンバ変数が宣

言されたデータ宣言部と、前記一連の処理において入力された前記設定項目ごとのデータを記憶するデータ記憶部とを備える記憶手段に前記入力データを記憶させ、該記憶手段に記憶された前記入力データに基づいて前記一連の処理を実行するウェブアプリケーションサーバの制御部に対して、

当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加させる情報付加処理と、

前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信させる情報送信処理と、

前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、

受信された返信情報に前記入力データが含まれていないかを検出させる返信情報検出処理と、

前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させる業務実行処理と

を実行させることを特徴とするウェブアプリケーションサーバ用支援プログラム。

【請求項 4】

前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、

前記制御部に、

設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、

前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、

前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させておいた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させ

ることを特徴とする請求項 2 又は請求項 3 に記載のウェブアプリケーションサーバ用支援プログラム。

【請求項 5】

前記情報記憶処理の後に、前記情報受信処理を行うために必要な情報を前記記憶手段に退避させて前記制御部に他の処理を実行させ、

前記情報受信処理により前記返信情報を受信したときに、前記退避された情報を復帰させて前記制御部に前記返信情報検出処理を再開させ

ることで前記ウェブアプリケーションサーバにおいて並列処理を実現させることを特徴とする請求項 1 乃至請求項 4 のいずれか 1 項に記載のウェブアプリケーションサーバ用支援プログラム。

【請求項 6】

請求項 1 ～請求項 5 に記載のウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたことを特徴とするウェブアプリケーションプログラム。

【請求項 7】

請求項 1 ～請求項 5 に記載のウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたことを特徴とするオペレーティングシステム。

【請求項 8】

通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前

記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される前記一連の処理の処理ステップに従って、前記返信情報に基づく処理を制御部が実行するサーバにおいて、

前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、

前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信する情報送信処理と、

前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、

受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出する返信情報検出処理と、

前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行し、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行する業務実行処理と

を前記制御部が実行することを特徴とするサーバ。

【請求項 9】

前記記憶手段は、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目毎のデータを記憶するデータ記憶部とを備え、

前記制御部は、

前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合に、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶し、

前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶する

ことを特徴とする請求項 8 に記載のサーバ。

【請求項 10】

クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信手段を用いてクライアントに送信し、該通信手段において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目ごとのデータを記憶するデータ記憶部とを備える記憶手段に前記入力データを記憶させ、該記憶手段に記憶された前記入力データに基づいて前記一連の処理を制御部が実行するサーバにおいて、

当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加する情報付加処理と、

前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信する情報送信処理と、

前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、

前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、

受信された返信情報に前記入力データが含まれていないかを検出する返信情報検出処理と、

前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶する業務実行処理と

を前記制御部が実行することを特徴とするサーバ。

【請求項 11】

前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、

前記制御部は、

設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を行うとともに、

前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、

前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させておいた識別子に基づいて区別して設定項目毎に前記データ記憶部に

記憶させ

ることを特徴とする請求項 9 又は請求項 10 に記載のサーバ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、クライアントにウェブ情報を送信するとともに、ウェブ情報に対するクライアントからの返信情報を受信して、返信情報に基づく一連の処理を実行するサーバとこのサーバに用いられる支援プログラムに関する。

【0002】

【従来の技術】

インターネット等のネットワークに接続されたウェブサーバに対して、クライアントからウェブ情報（例えば HTML 文書）の送信要求があった場合には、図 19 に示すように、ウェブサーバ 30 は、要求対象となるウェブ情報 31 を URL によって特定してクライアント 32 に送信することによって、クライアント 32 の要求に応じた情報提供を行っている。

【0003】

今日では、クライアント 32 による送信要求に応じてウェブ情報 31 をただ単に返信するだけではなく、図 20 に示すように、クライアント 32 からの要求に応じてある一連の処理を行うウェブアプリケーションサーバ 33 が数多く用いられている。ウェブアプリケーションサーバ 33 は、いわゆるフロントエンド（クライアント 32）とバックエンド（データベース管理システム等）との中間に位置する一種の支援プログラム（ミドルウェア）を備えており、この支援プログラムは、一連の処理を行うために必要な処理環境を整えて各種処理の流れを効率よく制御する役割を果たしている。また、ウェブアプリケーションサーバ 33 を用いることによって、ブラウザを備えているクライアント 32 であれば、ほとんどの処理をウェブアプリケーションサーバ 33 で処理させることができるので、クライアント 32 のハードウェア能力に依存せずに処理を行うことが可能となっている（例えば、非特許文献 1 参照）。

【0004】

ところで、既存のウェブアプリケーションサーバ33では、一連の処理に必要なHTML文書を生成するためのプログラム（以下、ビューとする。）が予め用意されており、ウェブアプリケーションサーバ33は、クライアント32のHTML文書の送信要求に応じて順次処理に必要なHTML文書を生成してクライアント32に送信し、そのHTML文書の入力フォームに基づいて入力された入力データをクライアント32から受信することによって、一連の処理に必要なデータを取得する。

【0005】

例えば、図21に示すような商品の発注処理を行うウェブアプリケーションサーバ33においては、クライアント32で注文入力を行うための注文入力画面を生成するビュー（以下、「ビューA」とする。）と、注文入力画面の入力フォームに対するクライアント32からの注文データに応じてその注文データの確認をクライアント32で行ってもらうための注文確認画面を生成するビュー（以下、「ビューB」とする。）と、注文確認画面における注文了解データに応じて注文処理が完了した旨の表示を行う注文完了画面を生成するビュー（以下、「ビューC」とする。）とが予め用意されている。ウェブアプリケーションサーバ33は、注文データや注文了解データをクライアント32から受信すると、注文内容（例えば、銘柄、数量、合計金額、発送先等）をデータベース34に記憶し、データベース34に記憶された注文内容に基づいて商品発送処理等を行っている。

【0006】

【非特許文献1】

日経BP社出版局編「デジタル用語辞典2002-2003年版」日経BP社発行、2002年3月18日第3版、P.431

【0007】

【発明が解決しようとする課題】

しかしながら、ウェブアプリケーションサーバ34では、ビューA～ビューCがネットワークを介していつでもアクセス可能な状態となっているので、クライアント32は発注処理の流れに関係なくビューA～ビューCにアクセスすることができる。図21に示した発注処理において処理がビューAからビューCまでの

手順に従って実際に行われているか否かは、クライアント 32 の送信要求に委ねられている。

【0008】

このため、図 22 に示すように、ビュー A に対するクライアント 32 からの応答に対して次の HTML 文書の送信要求先がビュー B となるようにビュー A には次の送信要求先が記憶されているが、その送信要求先がビュー B になるか否かはクライアント 32 の送信要求次第であり、ビュー A の次にビュー B が送信される予定となっても、場合によってはクライアント 32 がビュー B ではなくビュー C の送信要求を行ってしまい、ウェブアプリケーションサーバ 33 がビュー B により生成される HTML 文書ではなくビュー C により生成された HTML 文書をクライアント 32 に送信してしまうおそれがある。

【0009】

また、図 23 に示すように、クライアント 32 においてビュー B の注文確認用の HTML 文書が送信され、その後にビュー C により生成された発注処理が完了した旨の HTML 文書が送信され発注処理が完了した後に、クライアント 32 はブラウザの「戻る」ボタンを押すことによってキャッシュ内に記憶されていたビュー B の HTML 文書の画面を再度表示させることができる。クライアント 32 は、このようにキャッシュ内に記憶されていたビュー B の画面に従って再度ウェブアプリケーションサーバ 33 に注文了解データを返信してしまうおそれがあり、再度注文了解データをウェブアプリケーションサーバ 33 に返信した場合には二重に注文処理を行ってしまうという問題がある。

【0010】

さらに、図 24 に示すように、クライアント 32 がビュー A に対応する注文データをウェブアプリケーションサーバ 33 に返信する前に、悪意のある第三のクライアント 35 がクライアント 32 になりすまして他の注文データを送信してしまうおそれがある。この場合、従来のウェブアプリケーションサーバ 33 では、送信された注文データが、どのクライアントから送信されたどの HTML 文書に基づく返信であるかを判断することができないので、受信し注文データをクライアント 32 より送信された注文データであると判断して発注処理を進めてしまう

場合があり、クライアント32とウェブアプリケーションサーバ33との間で安全な発注処理を行うことが困難であるという問題があった。

【0011】

また、一般的なHTML文書においては、ビューA～ビューCのそれぞれのビュー内に、クライアント32が入力する必要のある入力項目が直接記述されているので、処理の変更等により入力項目が増減した場合には、各ビューの項目をそれぞれのビュー毎に変更しなければならず、変更処理に多大な労力を要するという問題があった。

【0012】

本発明は、上記の問題に鑑みて為されたものであり、クライアントにウェブ情報を送信するとともに、ウェブ情報に対するクライアントからの返信情報を受信して、返信情報に基づく一連の処理を実行するサーバにおいて、サーバが一連の処理の処理過程を把握することによって処理手順に沿わない処理がなされることを防止することができ、さらに、一連の処理に用いられるウェブ情報の内容を容易に変更することが可能なサーバ及びそのサーバに用いられる支援プログラムを提供することを課題とする。

【0013】

【課題を解決するための手段】

上記課題を解決するために、請求項1に係る発明は、通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される一連の処理の処理ステップに従って、前記返信情報に基づく処理を実行するウェブアプリケーションサーバの制御部に、前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信させる情報送信処理と、前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ス

テップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出させる返信情報検出処理と、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行させ、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行させる業務実行処理とを実行させるウェブアプリケーションサーバ用支援プログラムであることを特徴とする。

【0014】

請求項2に係る発明は、請求項1に記載のウェブアプリケーションサーバ用支援プログラムにおいて、前記記憶手段が、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目毎のデータを記憶するデータ記憶部とを備え、前記制御部に対して前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合には、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶させ、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶

手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させることを特徴とする。

【0015】

請求項3に係る発明は、クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信手段を用いてクライアントに送信し、該通信手段において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目ごとのデータを記憶するデータ記憶部とを備える記憶手段に前記入力データを記憶させ、該記憶手段に記憶された前記入力データに基づいて前記一連の処理を実行するウェブアプリケーションサーバの制御部に対して、当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加させる情報付加処理と、前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信させる情報送信処理と、前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段において受信させる情報受信処理と、受信された返信情報に前記入力データが含まれていないかを検出させる返信情報検出処理と、前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶させる業務実行処理とを実行させるウェブアプリケーションサーバ用支援プログラムであることを特徴とする。

【0016】

請求項4に係る発明は、請求項2又は請求項3に記載のウェブアプリケーションサーバ用支援プログラムにおいて、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される

入力データの設定項目の種類が重複する場合には、前記制御部に、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を実行させるとともに、前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させておいた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させることを特徴とする。

【0017】

請求項5に係る発明は、請求項1乃至請求項4のいずれか1項に記載のウェブアプリケーションサーバ用支援プログラムにおいて、前記情報記憶処理の後に、前記情報受信処理を行うために必要な情報を前記記憶手段に退避させて前記制御部に他の処理を実行させ、前記情報受信処理により前記返信情報を受信したときに、前記退避された情報を復帰させて前記制御部に前記返信情報検出処理を再開させることで前記ウェブアプリケーションサーバにおいて並列処理を実現させることを特徴とする。

【0018】

請求項6に係る発明は、請求項1～請求項5に記載のウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたウェブアプリケーションプログラムであることを特徴とする。

【0019】

請求項7に係る発明は、請求項1～請求項5に記載のウェブアプリケーションサーバ用支援プログラムにより実現される機能を備えたオペレーティングシステムであることを特徴とする。

【0020】

請求項 8 に係る発明は、通信手段を用いてクライアントにウェブ情報を送信し、該通信手段において前記ウェブ情報に対する前記クライアントからの返信情報を受信して該返信情報を記憶手段に記憶させ、前記記憶手段に記憶される前記一連の処理の処理ステップに従って、前記返信情報に基づく処理を制御部が実行するサーバにおいて、前記クライアントを特定するためのクライアント特定情報と前記ウェブ情報を特定するためのウェブ特定情報とを当該ウェブ情報に付加させる情報付加処理と、前記クライアント特定情報と前記ウェブ特定情報とが付加された前記ウェブ情報を特定の前記処理ステップにおいて前記通信手段を用いて前記クライアントに送信する情報送信処理と、前記ウェブ情報に付加された前記クライアント特定情報及び前記ウェブ特定情報と、前記ウェブ情報が前記クライアントに送信されたときの前記処理ステップに関する処理情報とを、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、受信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出する返信情報検出処理と、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報及び前記ウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一である場合に、前記記憶手段に記憶された前記処理情報より求められる処理ステップの次の処理ステップを前記記憶手段より求めて実行し、前記返信情報よりクライアント特定情報及びウェブ特定情報が検出され、かつ、検出された前記クライアント特定情報と前記記憶手段に記憶される前記クライアント特定情報とは同一であるが、前記返信情報より検出されたウェブ特定情報と前記記憶手段に記憶される前記ウェブ特定情報とが異なる場合には、前記記憶手段に記憶された前記処理情報より求められる処理ステップを再度実行する業務実行処理とを前記制御部が実行するサーバであることを特徴とする。

【0021】

請求項 9 に係る発明は、請求項 8 に記載のサーバにおいて、前記記憶手段が、

前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目毎のデータを記憶するデータ記憶部とを備え、前記制御部は、前記ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームが含まれている場合に、当該ウェブ情報を前記クライアントに送信する際に前記データ入力フォームにより入力される入力データの設定項目の種類を前記データ宣言部より判別し、前記情報記憶処理において前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに前記記憶手段に記憶し、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出された入力データの設定項目の種類と前記記憶手段に記憶される入力データの設定項目の種類とが同一である場合には、前記業務実行処理において、検出された前記入力データを設定項目毎に前記データ記憶部に記憶することを特徴とする。

【0022】

請求項10に係る発明は、クライアントにデータを入力させるためのデータ入力フォームが含まれるウェブ情報を通信手段を用いてクライアントに送信し、該通信手段において前記データ入力フォームに従って入力された入力データを返信情報として前記クライアントから受信し、前記一連の処理において使用される設定項目別のメンバ変数が宣言されたデータ宣言部と、前記一連の処理において入力された前記設定項目ごとのデータを記憶するデータ記憶部とを備える記憶手段に前記入力データを記憶させ、該記憶手段に記憶された前記入力データに基づいて前記一連の処理を制御部が実行するサーバにおいて、当該ウェブ情報に前記クライアントにデータを入力させるためのデータ入力フォームを付加する情報付加処理と、前記データ入力フォームが付加された前記ウェブ情報を前記通信手段を用いて前記クライアントに送信する情報送信処理と、前記データ入力フォームにより入力される入力データの設定項目の種類を前記記憶手段に記憶される前記データ宣言部より判別して、前記ウェブ情報が前記クライアントに送信される度に前記記憶手段に記憶させる情報記憶処理と、前記クライアントより返信される返信情報を、前記通信手段に受信させる情報受信処理と、受信された返信情報に前

記入入力データが含まれていないかを検出する返信情報検出処理と、前記返信情報より前記入力データが検出され、かつ検出された入力データの設定項目の種類と、前記記憶手段に記憶される設定項目の種類とが同一である場合には、検出された前記入力データを設定項目毎に前記データ記憶部に記憶する業務実行処理とを前記制御部が実行するサーバであることを特徴とする。

【0023】

請求項11に係る発明は、請求項9又は請求項10に記載のサーバにおいて、前記ウェブ情報に前記データ入力フォームが複数含まれている場合であって、前記データ入力フォームに従って入力される入力データの設定項目の種類が重複する場合には、前記制御部は、設定項目の種類が重複する入力データの識別を行うための識別子を前記データ入力フォーム毎に付して前記ウェブ情報に含ませる識別子追加処理を行うとともに、前記情報記憶処理において、前記クライアント特定情報、前記ウェブ特定情報、前記処理情報とともに、前記データ入力フォームにおいて入力される入力データの設定項目の種類を前記識別子に関連付けて前記記憶手段に記憶させ、前記業務実行処理において、前記返信情報より検出されたクライアント特定情報及びウェブ特定情報が前記記憶手段に記憶される前記クライアント特定情報及び前記ウェブ特定情報と同一であって、前記返信情報より検出される前記入力データの設定項目の種類と前記記憶手段に記憶される設定項目の種類とが同一識別子毎に一致する場合には、検出された前記入力データの設定項目を前記記憶手段に記憶させておいた識別子に基づいて区別して設定項目毎に前記データ記憶部に記憶させることを特徴とする。

【0024】

【発明の実施の形態】

以下、本発明の実施の形態を、図面を用いて説明する。

【0025】

図1は、本発明に係るウェブアプリケーションサーバのソフトウェア構成を示したブロック図である。このサーバ1は、Windows（登録商標）やLinux等のOS（operating system）2と、OS2上で動作するcgios3と、cgios3に基づいて実行されるアプリケーションソフトウェア4とによって

構成されている。

【0026】

c g i o s 3 は、c g i o s 用に記載されたアプリケーションソフトウェア 4 を実行するミドルウェアであり、アプリケーションソフトウェア 4 を従来の OS である OS 2 上で動かすためのエミュレータとしての役割を有している。また、c g i o s 3 は、ウェブサーバ用アプリケーションとしての機能及び CGI (Common Gateway Interface) 機能に加えて、後述する c g i o s 特有の機能を備えているので、ウェブアプリケーションサーバ用の支援ソフトウェア (ウェブアプリケーションサーバ用支援プログラム) としての役割も有している。この c g i o s 3 を用いることによって、c g i o s 3 において実現される処理を、プログラマーがアプリケーションソフトウェア 4 用のプログラムにおいて直接記述しなくても、c g i o s 3 が自動的に実行するので、プログラム作成労力の軽減を図ることが可能となる。

【0027】

図 2 は、サーバ 1 のハードウェア構成の概略を示したブロック図である。サーバ 1 は、クライアント 7 との間で各種データ (情報) の送受信を行う送受信部 8 と、CPU からなる制御部 9 と、ハードディスクからなる記憶部 10 及びデータベース 11 とを備えている。

【0028】

クライアント 7 は、パソコン、PDA (Personal Digital Assistance) 又は携帯電話等のブラウザを備えた通信機器である。送受信部 8 は NIC (Network Interface Card) 等により構成され、この送受信部 8 を用いることによって、サーバ 1 はネットワーク 12 を介してクライアント 7 との間でデータの送受信を行うことが可能となっている。ネットワーク 12 は外部に公開されるインターネット等のオープンなネットワークであっても、外部に公開されていない LAN 等のネットワークであってもよい。なお、図 2 においてはネットワーク 12 にはクライアント 7 が 1 台のみ接続されているが、複数接続されていてもかまわない。

【0029】

記憶部 10 には、c g i o s 3 用のプログラム言語により記述されたアプリケ

ーションソフトウェア4が記録されている。アプリケーションソフトウェア4は、ストリームプログラムとクラス定義プログラムとがコンパイルされることにより生成される。サーバ1は、そのアプリケーションソフトウェア4に基づいて、例えばクライアントから商品の注文を受け付ける受付処理のような一連の窓口処理（フロントエンドに対するバックエンド側のインターフェースに関する処理）を実行する。

【0030】

ここで、ストリームプログラムにはその窓口処理の全体的な流れが記述され、窓口処理の各段階における具体的な処理はクラス定義プログラムに記述されている。そして、このクラス定義プログラムがストリームプログラムに引用され、窓口処理が実行される。また、cgios3はクラス定義プログラムで使用されるメンバ変数をクラス定義プログラムのインスタンス（オブジェクト指向プログラム技術において一般的に使用されているインスタンスを意味し、クラスがオブジェクトの型を示すのに対して、インスタンスはオブジェクトの実体を示す。）ごとに管理する機能を有し、あるクラス定義プログラムのインスタンスからクライアント7に向けて送信されたメンバ変数が返信時にそのインスタンスのメンバ変数に戻るよう制御しているので（メンバ変数を発行するインスタンスと、メンバ変数を吸収するインスタンスとが同一）、ストリームプログラムがクラス定義プログラムに記載された処理命令を引用する場合には、ストリームプログラムにおいてメンバ変数を直接引数として記載する必要がない。したがって、引用される処理命令が異なるクラスに属するものであっても、また同一のクラスに属するものであっても、ストリームプログラムではそのクラスにおけるメンバ変数に左右されることなく処理手順を示すことが可能となっている。

【0031】

また、記憶部10には、窓口処理における処理の段階を示す処理ステップ情報や、次述するセッション値（クライアント特定情報）及び乱数値（ウェブ特定情報）が記憶（退避）される。ここで、処理ステップ情報とは、より具体的には制御部9が処理を行う際に参照するプログラムカウンタのアドレス値を記憶した情報である。さらに、記憶部10には、窓口処理においてクライアント7からデー

タが入力された際に、そのときの処理に関わるインスタンスごとに入力データを設定項目として分けて記憶するデータ記憶部が設けられている。

【0032】

データベース11には、窓口処理において処理が確定した入力データが入力される。データベース11は、この点において、窓口処理において入力された入力データを一時的に保存するデータ記憶部と相違する。データベース11に記憶される設定項目は、クラス定義プログラムで宣言されるメンバ変数の設定項目と同期されており、クラス定義プログラムにおいて設定項目が変更されると、cgios3が自動的に同期される設定項目を変更する。

【0033】

制御部9は、サーバ1を構成するコンピュータのCPUであり、アプリケーションソフトウェア4に従って各種処理を行うほか、cgios3のカーネル(cgiosの基本機能を実現する部分)の指示に従ってcgios3特有の機能を実現させるための処理を実行する。以下、cgios3の指示に基づいて制御部9によって実行されるcgios3特有の機能を説明する。

【0034】

図3は、サーバ1からクライアント7に送信されるHTML文書(ウェブ情報)及びクライアント7からサーバ1に返信される返信情報との送受信関係を示したデータフロー図である。

【0035】

クライアント7によって取引開始ページのHTML文書の送信要求(取引開始ページのURLへのアクセス要求)がなされた場合、制御部9は、セッション値と乱数値と生成する。ここで、セッション値とは、サーバ1が特定のクライアント7のために行う一連の窓口処理を識別するために用いられる値であり、制御部9は、このセッション値を用いることによってその窓口処理の内容の特定及びクライアント7の特定を行う。また、乱数値は、サーバ1がクライアント7に対して送信するHTML文書を特定するために用いられるものである。なお、本実施例ではHTML文書の特定を行うために乱数値を用いることとしたが、HTML文書の特定を行うことが可能な識別情報としての機能を有するものであるならば

乱数値に限定されるものではない。また、サーバ1は、ストリームプログラムに記述されている第1の処理ステップにおける処理命令をクラス定義プログラムから引用することによって第1の処理ステップに用いられるHTML文書（以下、第1のHTML文書とする。）を生成する。

【0036】

その後、制御部9は、生成したセッション値と乱数値とを暗号化し、暗号化されたセッション値と乱数値とをHTMLのHIDDENパラメータを用いて第1のHTML文書に付加して送受信部よりクライアント7に送信する。また、制御部9は、第1のHTML文書の送信と同時に、セッション値、乱数値、処理ステップ情報を記憶部10に記憶（退避）し、さらに、第1のHTML文書にクライアント7にデータを入力させるための入力フォームが設けられている場合には、入力フォームにより入力することが可能な設定項目情報をインスタンス毎に記憶する。その後、制御部9は窓口処理から解放される。

【0037】

サーバ1より送信された第1のHTML文書がクライアント7において受信されると、第1のHTML文書がクライアント7のウェブブラウザ画面上に表示される。クライアント7は、第1のHTML文書の入力フォームに従ってデータ（入力データ）を入力することができ、入力処理を行った後に「次へ」等のボタンをクリックすることによって、取引開始ページのURLへのアクセス要求を再度行うとともに、入力処理が行われた入力データと、暗号化されたセッション値及び乱数値とを返信情報としてサーバ1に返信する。なお、サーバ1のcgi os 3は、セッション値及び乱数値に基づいてクライアント及びHTML文書の特定を行っているため、クライアント7からサーバ1へのアクセス要求が行われるURLは、取引開始ページのURLだけである。この点において、サーバ1は、窓口処理の処理内容毎に異なるURLをアクセス要求先として指定する従来のウェブアプリケーションサーバと相違する。

【0038】

返信情報がサーバ1の送受信部8に返信されると、制御部9は、一連の窓口処理を再開する。制御部9では、返信情報に暗号化されたセッション値および乱数

値が含まれているか否かの検出を行い、セッション値及び乱数値が含まれていない場合には、受信した返信情報はサーバ1が送信したHTML文書とは無関係に返信された情報であると判断する。

【0039】

返信情報より暗号化されたセッション値および乱数値が検出された場合には、制御部9は、暗号化されたセッション値及び乱数値を抽出して復号化するとともに、記憶部10に記憶されていたセッション値及び乱数値を読み出して、復号化されたセッション値及び乱数値を、記憶部10に記憶されているセッション値及び乱数値と比較する。制御部9は、復号化されたセッション値と記憶部10に記憶されているセッション値とが同一である場合には、HTML文書を送信したクライアントと、返信情報を返信したクライアントとが同一クライアントであると判断する。一方、比較されるセッション値が異なる場合、制御部9はクライアントが異なるものと判断して受信した返信情報を破棄する。

【0040】

比較されるセッション値が同一である場合、制御部9は、さらに復号化された乱数値と第1のHTML文書を送信したときに記憶部10に記憶された乱数値とが同一であるか否かを比較し、乱数値が同一である場合には、第1のHTML文書の入力フォームに従ってクライアント7が入力した入力データが返信情報として返信されたものであると判断し、乱数値が異なる場合には第1のHTML文書とは異なるHTML文書に基づく処理に対する返信情報であると判断する。

【0041】

セッション値及び乱数値を比較することによって、制御部9が、受信した返信情報が第1のHTML文書の入力フォームに基づく返信情報であると判断した場合には、記憶部10に記憶されているクラス定義プログラムにおけるインスタンス毎の設定項目情報と、返信情報のインスタンス毎の入力データの設定項目情報とを比較し、記憶部10に記憶されている設定項目と一致する入力データを、そのインスタンスにおいて用いられる設定項目毎に区分けしてデータ記憶部に記憶する。このように、クライアント7によって入力された入力データをクラス定義プログラムに用いられるインスタンスのメンバ変数の設定項目毎にデータ記憶部に

記憶しておくことによって、ストリームプログラムがその後の処理ステップにおいてクラス定義プログラムに記載された処理命令を引用する場合には、データ記憶部に記憶されたインスタンス毎の入力データを設定項目毎の既設値としてその後の処理に利用される。記憶部10に記憶されている設定項目と一致しない入力データは破棄される。その後、制御部9は、記憶部10に記憶された処理ステップ情報より、ストリームプログラムに示された一連の窓口処理においてどの処理ステップまで処理が終了しているかを判断し、判断される処理ステップの次の処理ステップ、つまり第2の処理ステップの処理を実行する。

【0042】

第2の処理ステップにおいて、制御部9は、ストリームプログラムの第2の処理ステップで引用されている処理命令を実行することによって第2の処理ステップに用いられるHTML文書（以下、第2のHTML文書とする。）を生成する。その後、制御部9は、第2のHTML文書であることを特定するための他の乱数値を生成し、第1の処理ステップにおいて生成したセッション値とともに暗号化して第2のHTML文書に付加する。

【0043】

制御部9は、第2のHTML文書を、送受信部8を介してクライアント7に送信し、さらに、第2のHTML文書の送信と同時に、セッション値、乱数値、処理ステップ情報、第2のHTML文書の入力フォームに使用されたクラス定義プログラム毎の設定項目情報を記憶部10に記憶（退避）して、第1の処理ステップにおいて説明した処理と同様の処理を繰り返す。

【0044】

第1の処理ステップにおいて返信された返信情報に、暗号化されたセッション値および乱数値が含まれている場合であって、暗号化されたセッション値と記憶部10に記憶されているセッション値は同一であるが乱数値が異なる場合には、第1のHTML文書に対してクライアント7が返信情報を返信したものではないと判断できるので、記憶部10に記憶される処理ステップ情報により特定される処理を再度繰り返して実行する。再度処理を繰り返して実行する場合には、制御部9は、受信された返信情報を破棄し、再度第1のHTML文書を作成して前述

のセッション値と新たに作成した第1のHTML文書を特定するための新たな乱数値とを暗号化して第1のHTML文書に付加する。その後、制御部9は送受信部10を介して第1のHTML文書を再度クライアント7に送信する。再度第1のHTML文書を送付する場合に、制御部9は、記憶部10に記憶されている乱数値のみを変更して再度記憶部10にセッション値、処理ステップ情報等の各種情報を記憶（退避）する。

【0045】

制御部9は、上記の処理を繰り返して、ストリームプログラムに記憶されている処理ステップを順次実行し、一連の処理の終わりに記憶部10のデータ記憶部に更新記憶された入力データを、一連の窓口処理における確定したデータとしてデータベース11に登録する処理を行った後に、窓口処理が終了した旨のHTML文書を作成してクライアント7に送信して処理を終了する。

【0046】

次に、簡単なストリームプログラム及びクラス定義プログラムを用いて、より具体的にサーバ1とクライアント7との間のデータの送受信方法を説明する。

【0047】

図4～図7は牡蛎料理の注文処理における実行プログラムを示したプログラムリストであり、図4はストリームプログラム、図5、6は、牡蛎料理の注文に関するクラス定義プログラム、図7は、図4に示したストリームプログラムに従って作成されるHTML文書のページ定義を示したプログラムである。

【0048】

牡蛎料理の注文処理は、基本的に5つの処理ステップによって完了する。ステップ1は、使用されるクラスの宣言及び生成されたインスタンスで用いられる設定項目の初期値設定を行う処理ステップである。

【0049】

ステップ2は、クライアントに牡蛎の銘柄、調理方法、添える調味料の3種類の設定項目のデータの入力と、コメントの入力を行ってもらう処理ステップであり、この処理を行うために注文入力画面（以下、ページA1とする。）のHTML文書がサーバ1からクライアント7に送信される。図8及び図9はページA1

に該当する画面表示を示したものである。

【0050】

ステップ3は、クライアント7において入力された牡蛎の銘柄、調理方法、添える調味料、コメントの内容を確認してもらう処理ステップである。この処理を行うために、ステップ1においてクライアント7によって入力された注文内容を確認表示させる注文確認画面（以下、ページB1とする。）のHTML文書がサーバ1からクライアント7に送信される。図10及び図11がページB1に該当する画面表示を示したものである。なお、ステップ3では、クライアント7により入力された調味料の種類が2種類以下であるか否かを検査するエラー処理が実行される。3種類以上の場合には、図12に示すようなエラーメッセージが画面上に表示される。

【0051】

ステップ4は、注文の確認が為された設定値情報をデータベース11に記憶する処理である。

【0052】

ステップ5は、ステップ3で確認された内容により牡蛎料理の注文を承った旨を知らせる処理ステップであり、この処理を行うためにステップ3で確認された注文内容と注文処理が完了した旨とを表示する注文完了画面（以下、ページC1とする）のHTML文書がサーバ1からクライアント7に送信される。図13がページC1に該当する画面表示である。

【0053】

また、図7（a）に示したプログラムは、ページA1のHTML文書を作成するためのページ定義を示したものであり、（b）に示したプログラムは、ページB1のHTML文書を作成するためのページ定義を示したものであり、（c）に示したプログラムは、ページC1のHTML文書を作成するためのページ定義を示したものである。図7（a）～（c）に示したプログラムは、HTML（HyperText Markup Language）で記載されており、<h3>～</h3>はHTML文書の表示画面上に見出しを表示させるためのタグである。また、<input type="submit" action="next" label="次へ">は、「次へ」と書かれたボタンを表示

させるタグであり、クライアント7でこのボタンがクリックされた場合には、返信情報として” next” という戻り値がサーバ1に返信される。＜#slot＞は、ストリームプログラムの所定の処理ステップにおいて呼び出されたクラス定義プログラムの処理命令に応じてその表示内容が変化することを示したものである。

【0054】

図5、6に示したクラス定義プログラムを説明する。このクラス定義プログラムは、牡蛎料理の注文に用いられる設定項目ごとのメンバ変数の宣言と、その設定項目情報を用いて処理される処理命令とをまとめたものであり、牡蛎料理の注文に関する処理の際に必要な情報が集約されている。図5、6に示したクラス定義プログラムにおいては、package OysterOrder;に示されるようにOysterOrderというクラスが定義されている。

【0055】

クラス定義プログラムは、#BEGIN#DECLARE～#END#DECLAREで挟まれた部分で示された設定項目のデータ宣言部p1と、sub ～ { ～～ } で示された処理命令部p2～p7とで構成されている。

【0056】

データ宣言部p1では、データベース11に対応する設定項目毎のメンバ変数の宣言を行っており、注文ID、牡蛎の銘柄を示すコード、調理方法を示すコード、調味料コードを入力するための符号付整数型のメンバ変数宣言と、調理方法についてのコメントを入力するためのテキスト入力用のメンバ変数宣言とが行われている。クラス定義プログラムがコンパイルされると、コンパイルの度に宣言されたメンバ変数に対応する設定項目がデータベースに作成される。このため、クラス定義プログラムにおいてメンバ変数の設定項目が変更された場合には、クラス定義プログラムのコンパイルの際に自動的にデータベース11の設定項目も変更される。

【0057】

処理命令部p2は、クラスで用いられる設定項目毎の記憶領域を、記憶部10のデータ記憶部に確保することによりインスタンスを生成する処理命令である。

【0058】

処理司令部 p 3 は、データ記憶部に記憶された入力データを、データベース 11 に登録する際に用いるシーケンス番号を生成する処理命令であり、この処理命令を実行させることによって、データベース 11 にデータが登録されるときに既に登録されているデータ上に上書き保存されることを防止することが可能となる。

【0059】

処理司令部 p 4 は、インスタンスで使用する設定項目の初期化を行う処理命令である。この処理命令がストリームプログラムにおいて引用されて実行されると、牡蛎の銘柄の銘柄コードを入力するためのインスタンスのメンバ変数 {brand} に岩牡蛎を示す設定値 1 が代入され、調理方法コードを入力するためのメンバ変数 {recipe} に生食を示す設定値 1 が代入され、調味料コードを入力するためのメンバ変数 {trimming} にレモンのみを示す設定値 2 が代入され、調理方法のコメントを入力するメンバ変数 {comment} が空欄に設定される。

【0060】

処理司令部 p 5 は、ページ A 1 に設けられる注文入力フォームを作成するための処理命令である。この処理命令がストリームプログラムで引用されて実行されると、図 8 に示すような HTML 文書（ページ A 1）が作成される。cgios 3 の指示に従って制御部 9 は、作成されたページ A 1 に従ってクライアント 7 がデータを入力することが可能な設定項目を、ページ A 1 を作成する際に引用されたインスタンスに関連づけて記憶部 10 に記憶する。クライアント 7 が、ページ A 1 に基づいて牡蛎の銘柄の種類等の入力データ 20 の設定を行ってから、画面上の「次へ」ボタン 21 をクリックすると、サーバ 1 に対して返信情報が送信される。返信情報が送信されたサーバ 1 では、返信情報に含まれる入力データの設定項目が記憶部 10 に記憶される設定項目に該当するか否かの判断を制御部 9 が行い、該当する設定項目がある場合には、記憶部 10 のデータ記憶部に記憶されている設定項目毎の入力データを返信情報に含まれる入力データへ自動的に更新し、該当しない場合には返信情報に含まれる入力データを破棄する。

【0061】

処理司令部 p 6 は、返信情報に調味料コードに関連する設定項目の入力データ

が含まれており、さらに記憶部10にも調味料コードに関連する設定項目が記憶されている場合であって、互いの設定項目は一致するが、入力コードのデータ値が許容データ値でない場合、例えば、調味料コードが同時に3種類選択された場合に、ページA1上にエラーメッセージを表示させる処理命令である。

【0062】

処理命令部p7は、ページB1において表示される注文確認項目と、ページC1において表示される注文完了項目とを作成する処理命令である。ページB1とページC1とは画面最上部に表示される見出しの表示と画面の最下部に表示されるボタンの種類が異なるだけで、牡蛎の銘柄、調理方法、調味料、コメントからなる表示項目は同じであるため、ステップ2における注文確認画面及びステップ4における注文完了画面の両方のHTML文書を作成する際に用いられる。

【0063】

次にストリームプログラムについて説明する。ストリームプログラムは、クラス定義プログラムに記載された処理命令を引用することによって一連の窓口処理の流れが記述されたプログラムである。

【0064】

図4に示すストリームプログラムは、top、first、second、third、fourthの5つの処理ステップによって構成されており、この処理ステップの情報が、処理ステップ情報として記憶部10に記憶される。

【0065】

まずustream input#order(\$o) [# 注文入力ストリーム ~~]において、このストリームプログラムが注文入力ストリームであることを説明するとともに、この注文入力ストリームにおいて使用されるクラスが\$oで示される一つのインスタンスだけであることを宣言している。

【0066】

```
top:$o = new <APP>::OysterOrder($S);
```

より、new処理命令（処理命令部p2）が実行されることによってOysterOrderで示されるクラス用の記憶領域が確保され、その記憶領域に\$oで示されるインスタンスが生成される。さらに、\$o->default;によって処理命令default（処理命

令部 p 4) が実行されて、\$oで示されるインスタンスの設定項目の初期値設定が行われる。

【0067】

なお、このストリームプログラムにおいては、1つのインスタンス\$oしか使用されていないが、複数のインスタンスを用いて一連の処理を行う場合には、

```
stream input#order($o1,$o2) [ # 注文入力ストリーム
```

```
top:
```

```
    $o1 = new <APP>::OysterOrder($S);
```

```
    $o1->default;
```

```
    $o2 = new <APP>::OysterOrder($S);
```

```
    $o2->default;
```

に示すように同一のクラスを用いて複数のインスタンス\$o1、\$o2を宣言することも可能である。

【0068】

次の処理ステップfirstでは、図7(a)に示したページ定義を利用して、図8に示すページA1が作成される。ここで、\$V->[SLOT] = [slot => \$o->input];と記載されていることから、図7(a)の<#slot>のタグの部分に、インスタンス\$oのinput処理命令(処理命令部p5)によって作成される注文入力フォームが、はめ込まれるようにして表示される。処理ステップfirstのプログラムには記載されていないが、firstの処理ステップにより作成されたページA1には、セッション値と乱数とが暗号化されて付加され、さらに、記憶部10にセッション値、乱数値、処理ステップ情報(ここでは処理ステップがfirstである旨の情報)が記憶されるとともに、input処理命令において使用される設定項目情報、具体的には、牡蛎の銘柄、牡蛎の料理方法、添える調味料、コメントの4つの設定項目情報がインスタンス\$oに関連付けて記憶される。また、

```
if ($V->[action] eq "next") [ goto second; ]
```

```
goto first;
```

より、ページA1において図8に示される「次へ」ボタン21がクリックされると、ストリームプログラムにおける次の処理ステップsecondに制御部9の処理が

移行する。

【0069】

なお、ページA1の「次へ」ボタン21がクリックされると処理ステップsecondに制御部9の処理が移行することとなるが、処理が移行される前に、既に説明したように、制御部9がクライアント7からの返信情報を取得して、セッション値、乱数値、入力データにおける設定項目情報の抽出・比較処理を行う。制御部9は、返信情報よりセッション値及び乱数値が検出されない場合及びセッション値が一致しない場合には、返信情報を破棄する。また、検出されたセッション値と記憶部10に記憶されたセッション値は同一であるが、検出される乱数値と記憶部10に記憶される乱数値が異なる場合には、クライアント7においてブラウザの「戻る」ボタン22等を押してキャッシュに記憶されている古いHTML文書を表示させた後に、その古いHTML文書に基づいて返信情報が返信されたものと判断することができる。古いHTML文書に基づいて返信情報が返信されたものと判断された場合、制御部9は再度処理ステップfirstの処理を実行し、検出されたセッション値及び乱数値が記憶部10に記憶されるセッション値及び乱数値と一致する場合には、返信情報よりクライアント7が入力した入力データの設定項目を検出して、データ記憶部の設定項目に、該当する入力データを更新記憶させる。該当しない項目の入力データに関してはそのデータを破棄してから、次の処理ステップsecondに処理を移行する。

【0070】

次の処理ステップsecondでは、ページBを作成する前に、制御部9 返信情報の内容についてエラーがないかチェックする。

【0071】

```
if ($o->[ERROR] = $o->check) [ goto first; ]
```

によって、インスタンス\$oのcheck処理命令（処理命令部分p6）の処理が実行され、クライアント7において調味料が3種類選択された場合には、処理ステップfirstに戻って、図12に示すようにエラー表示を表示させる。

【0072】

処理ステップsecondにおいて、調味料が図8に示すように1種類しか選択され

ていない場合には、図 7 (b) に示したページ定義を利用して、図 10 に示したページ B 1 が作成される。また、図 9 に示すように、調味料の選択が 2 種類以下である場合には、図 11 に示したページ B 1 が作成される。

【0073】

このページ B 1 の作成の際には、インスタンス \$o の print 処理命令 (処理命令部 p 7) の処理が実行され、処理ステップ first において返信情報としてクライアント 7 からサーバ 1 に返信されたページ A 1 の入力データが表示される。ページ B 1 においては、ページ A 1 と違って「注文実行」ボタンと、「戻る」ボタンとが表示される。ここで「注文実行」ボタンがクリックされると次の処理ステップである third に処理が進み、「戻る」ボタンをクリックすると一つ前の処理ステップである first の処理に戻ることとなる。

【0074】

次の処理ステップ third では、インスタンス \$o に用いられるメンバ変数 {orid} に対して nextorid 処理命令 (処理命令部 p 3) を実行することによってシーケンス番号を代入し、\$\$S[DA]->saveObject(\$o); により、シーケンス番号が入力されたメンバ変数 {orid} を主キーとしてインスタンス \$o の設定項目毎の入力データがデータベース 11 に記憶される。

【0075】

その後、処理ステップが fourth に進んで、図 7 (c) に示したページ定義を利用して、図 13 に示したページ C 1 が作成される。処理ステップ fourth により一通りの牡蛎料理の注文処理が終了するが、ビュー C 1 に表示される「最初に戻る」ボタンをクリックすることによって、再度処理ステップ first の処理を繰り返す行うことが可能である。

【0076】

以上説明したように本発明に係るサーバ 1 は、サーバ 1 が実行する一連の窓口処理をサーバ 1 でその処理ステップの処理状況を確認しながら実行することが可能となる。このため、一連の窓口処理が手順に従って行われるか否かがクライアント 7 に委ねられていた従来のウェブアプリケーションサーバに較べて、本発明に係るサーバ 1 は、一連の窓口処理が処理手順に従って実行されるように、サー

バ7に送信するHTML文書の表示内容を制御することができ、サーバ1側の裁量で窓口処理を進めることによって意図しない処理の遷移を防止することが可能となる。

【0077】

また、実行プログラムを作成するプログラマーはストリームプログラムを作成するだけで、クライアントの特定及び処理ステップ毎の返信情報の確認等の処理をプログラム上に記述しなくても、クライアントの特定、返信情報における入力データの保存等が確実に実行されるので、プログラム作成の労力が低減されるとともに、プログラムの修正・変更処理が容易になる。

【0078】

さらに、本発明に係るサーバ1は、cgios3の指示により、返信情報に含まれる入力データの設定項目の種類が、送信したHTML文書の入力フォームにおいて入力可能な設定項目の種類に対応するものであるか否かを制御部9が判断して、入力データの設定項目と記憶されたが設定項目とが異なる場合にはその入力データを破棄し、一致する場合にはその入力データをインスタンスに関連付けて自動的に記憶部10のデータ記憶部に記憶させることができる。このため、本発明に係るサーバ1において、意図しない入力データがクライアント7からの返信情報として記憶されてその後の処理に反映されることを防止することができる。

【0079】

また、プログラマーは、返信情報に含まれる入力データが適正なデータであるか否かの判断を行う処理をプログラムで規定しておく必要がないので、プログラム作成労力の一層の低減を図ることが可能となるとともに、サーバ1の安定性向上を図ることが容易となる。

【0080】

さらに、HTML文書の送信時及び返信情報の受信時にのみ、記憶部10に記憶された必要なデータを読み出して（復帰させて）制御部9に必要な処理を行わせることができるので、制御部9は複数のストリームプログラムを用いたマルチタスク処理を容易に実現することが可能となる。

【0081】

また、クラス定義プログラムごとの一連の窓口処理で使用される設定項目毎のメンバ変数とその設定項目を利用する処理命令とがまとめられており、他のクラス定義プログラムやストリームプログラムにおいてメンバ変数は記載されないの
で、一連の窓口処理で使用されるメンバ変数が変更される場合であっても、クラス定義プログラム内のメンバ変数の宣言内容を変更するだけで良い。このため、他のクラス定義プログラム及びストリームプログラム等を変更することなく、クラス定義プログラムに記載されるメンバ変数の設定項目や処理命令を変更するだけで容易に仕様変更を行うことが可能となり柔軟性に富んだシステム構築を行うことが可能となる。特に、cgios3では、同一ストリームプログラムにおいて同一のクラスから生成された複数のインスタンスが用いられている場合であっても、cgios3がクラスのメンバ変数を、各インスタンスの設定項目毎に管理するので、プログラマーはストリームにおいて使用されるメンバ変数の設定項目が重複する場合であっても、プログラム上でその違いを明確にすることなくストリームプログラムを作成することができる。

【0082】

例えば、図14に示すように、1画面において牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメントの項目を2組表示させるページDを作成する場合、ストリームプログラムにおいては、\$o1と\$o2との2つのインスタンスを宣言することによって、OysterOrderクラスに使用されるメンバ変数の設定項目毎のデータを、\$o1と\$o2とからなる識別子を用いてそれぞれを区別し、別々に記憶する。ビューDをクライアント7に送信する場合には、入力フォームにおけるインスタンス\$o1の設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とインスタンス\$o2の設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とを区別して記憶部10に記憶させ、返信情報より入力データを検出する場合も、インスタンス\$o1の入力データの設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とインスタンス\$o2の入力データの設定項目（牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメント）とを区別して抽出し、それぞれデータ記憶部に別々の入力データとして記憶する。

【0083】

このように、重複する設定項目からなる入力フォームをHTML文書で作成する場合であっても、各設定項目に対応する処理命令を1つ用意して、同一クラスを用いて複数のインスタンスを生成することによって、各設定項目の区分けをc g i o sで自動的に行わせることができ、クラス定義プログラムの処理命令だけを変更するだけで容易に入力フォームの表示形式、設定項目等の仕様変更を容易に行うことができる。さらに、クラス定義プログラムで使用されるメンバ変数は他のクラス定義プログラムやストリームプログラム等に依存しないので、クラス定義プログラムの設定項目等を変更しても、その変更に起因するシステム全体のエラーの発生を低減させることができる。

【0084】

なお、c g i o s 3は、一連の窓口処理の流れを示すストリームプログラムと、このストリームプログラムで使用されるインスタンスのメンバ変数とそのメンバ変数を用いる処理命令とでまとめたクラス定義プログラムとを組み合わせるプログラム構成を行うので、今までのHTML文書の作成技法では困難であったと入力フォームを容易に作成することが可能となる。例えば、注文されたデータの一覧表をHTML文書において作成する場合には、表外枠、見出し、各種設定項目等をインスタンスの処理命令により特定の出力形式で作成し、作成された一覧表をストリームプログラムにおいて引用することによって、一覧表をHTML文書に挿入するようにして貼り付けることが可能となるので、各行が入力フォームを持つような表一覧を簡単に作成することができる。さらに、ストリームプログラムから他のストリームプログラムを呼び出すことも可能であるため、上述の牡蠣料理の注文処理を示したストリームプログラムにおいて、表一覧を作成する表一覧作成ストリームを呼び出し、この表一覧作成ストリームにおいてインスタンスから表一覧を作成するための処理命令を引用して表一覧を作成し、表一覧作成後に牡蠣調理の注文処理のストリームに処理を戻すことも可能である。さらに、表一覧を作成する方法を応用することによって、HTML文書における表一覧の他に、Microsoft excel（登録商標）に対して一覧表形式にデータを出力することも可能である。

【0085】

また、ストリームプログラムにおいて他のストリームプログラムを呼び出す方法とHTML文書を作成するためのページ定義とを活用することによってウェブページにおけるウィンド管理をサーバ1で行うことが可能となる。従来のウェブアプリケーションでは、基本的に1つのウェブページでは1つの処理しか実行することができず、複数の処理を行う場合にはクライアント7が使用するブラウザのフレーム機能を用いることが一般的であった。しかしながらブラウザのフレーム機能の動作は、基本的にクライアント7の裁量に委ねられているので、サーバ1が意図する形式のウィンド表示とならない場合があった。cgios3では、HTML文書のページ定義によりウィンド表示形式を決定し、各ウィンド内で個別にストリームプログラムを実行させて、各ストリームプログラムの実行結果として出力されるHTML文書を該当するウィンドに貼り付けて1つのウェブページを生成することができるので、ブラウザのフレーム機能に依存することなくウィンド管理を行うことが可能となる。

【0086】

さらに、ウィンド毎に実行される各ストリームプログラムの処理に応じて記憶部10に記憶されたデータの読み出し（復帰）、記憶（退避）を行うことができるので、マルチウィンドを用いたマルチタスク処理を簡易に実現することが可能となる。

【0087】

図15に示すような3つのウィンドを有するHTML文書をcgios3で作成する方法を、図16に示したストリームプログラムと、図17に示したページ定義とを用いて説明する。まず、図16(a)に示した起動ストリームプログラムにより、ウィンドの設定がされていないウェブページ（起動ストリームプログラムのp#fork(" ", "/~/", "this")の"this"がその旨を示している）上で図16(b)に示すウィンド起動ストリームプログラムを実行し、p#wait()により起動ストリームプログラムの処理の待機を行う。

【0088】

起動ストリームプログラムにおいてウィンド起動ストリームプログラムが実行

されると、図 17 (a) に示したページ定義が用いられた図 15 に示すような 3 つのウィンドが形成され、このウィンドの”menu”部分において図 16 (c) に示すメニューストリームプログラムが実行される。

【0089】

メニューストリームプログラムが実行されて、ウィンドの” menu” 部分に図 17 (b) に示したページ定義のメニュー表示が表示され、メニュー表示において注文入力をクリックされるとウィンドのview部分において牡蛎料理の注文処理に関するストリームプログラムが実行されて、図 18 に示すように牡蛎料理の注文に関する入力フォームが作成される。また、メニュー表示において注文一覧表示をクリックされると、注文一覧表示を行うストリームプログラムが実行される。

【0090】

このように、複数のページ定義と、複数のストリームプログラムを活用することによって、一連の窓口処理の内容に応じて入力フォーム等を表示させるウィンドを使い分けることが可能となる。さらに、クライアント 7 のブラウザにおけるフレーム機能に依存されずにウィンド管理を行うことができるので、サーバ 1 側で意図したウィンド表示をクライアント 7 のウェブページ上で表示させることが可能となる。

【0091】

以上本発明に係るサーバを説明してきたが、本発明に係るサーバは上記発明の実施の形態に示したものに限定されるものではない。例えば、本発明の実施の形態においては、c g i o s 3 が OS 上で動くミドルウェアであるものとして説明を行ったが、ウェブサーバ用アプリケーション（例えばApache等）において実行される支援ソフトウェア（支援プログラム）であっても良い。さらに、c g i o s 3 自体がオペレーションシステムであって、オペレーションシステムにおいて c g i o s 特有の機能を実現させたものであっても良い。

【0092】

【発明の効果】

以上説明したように本発明を用いることによって、サーバが実行する一連の処理をサーバでその処理ステップの処理状況を確認しながら実行することが可能と

なる。

【0093】

また、本発明では、返信情報に含まれる入力データの設定項目の種類が、送信したウェブ情報の入力フォームにおいて入力可能な設定項目の種類に対応するものであるか否かを制御部が判断して、入力データの設定項目と記憶されたが設定項目とが一致する場合にはその入力データをデータ宣言部に関連付けて自動的にデータ記憶部に記憶させることができるので、意図しない入力データがクライアントからの返信情報としてデータ記憶部に記憶されてその後の処理に反映されることを防止することができる。

【0094】

さらに、ウェブ情報の送信時及び返信情報の受信時にのみ、記憶部に記憶された必要なデータを読み出して（復帰させて）制御部に必要な処理を行わせることができるので、制御部は複数の一連の処理を同時に実行するマルチタスク環境を容易に実現することが可能となる。

【図面の簡単な説明】

【図1】

本発明に係るサーバのソフトウェア構成を示したブロック図である。

【図2】

本発明に係るサーバのハードウェア構成を示したブロック図である。

【図3】

本発明に係るサーバとクライアントとのデータの送受信関係を示したデータフロー図である。

【図4】

牡蛎料理の注文処理の処理手順を示したストリームプログラムのプログラムリストを示した図である。

【図5】

牡蛎料理の注文処理に用いられるクラス定義プログラムのプログラムリストを示した第1の図である。

【図6】

牡蛎料理の注文処理に用いられるクラス定義プログラムのプログラムリストを示した第2の図である。

【図7】

牡蛎料理の注文処理に用いられるHTML文書のデータ定義示した図である。

【図8】

牡蛎料理の注文処理のステップ2において使用されるHTML文書の表示画面を示した第1の図である。

【図9】

牡蛎料理の注文処理のステップ2において使用されるHTML文書の表示画面を示した第2の図である。

【図10】

牡蛎料理の注文処理のステップ3において使用されるHTML文書の表示画面を示した第1の図である。

【図11】

牡蛎料理の注文処理のステップ3において使用されるHTML文書の表示画面を示した第2の図である。

【図12】

牡蛎料理の注文処理における調味料の種類に関するエラー表示が為されたHTML文書の表示画面を示した図である。

【図13】

牡蛎料理の注文処理のステップ5において使用されるHTML文書の表示画面を示した図である。

【図14】

牡蛎料理の注文処理において牡蛎の銘柄、牡蛎の調理方法、添える調味料、コメントの項目が2組表示されたHTML文書の表示画面を示した図である。

【図15】

3つのウィンドを有するHTML文書の表示画面の概略を示した図である。

【図16】

3つのウィンドを有するHTML文書を作成するためのストリームプログラム

のプログラムリストを示した図である。

【図 17】

3つのウィンドを有するHTML文書を作成するために用いられるページ定義を示した図である。

【図 18】

3つのウィンドを有するHTML文書の表示画面を示した図である。

【図 19】

ウェブサーバとクライアントとの間におけるウェブ情報の送受信方法を示した概略図である。

【図 20】

従来のウェブアプリケーションサーバとクライアントとの間におけるウェブ情報の送受信方法を示した概略図である。

【図 21】

ウェブアプリケーションサーバとクライアントとの間における商品の発注処理の方法を示した概略図である。

【図 22】

ウェブアプリケーションサーバに対してクライアントが発注処理の流れに関係なくウェブ情報の送信要求を行ってしまう場合を示した図である。

【図 23】

ウェブアプリケーションサーバに対してクライアントが二重に発注処理をしてしまう場合を示した図である。

【図 24】

ウェブアプリケーションサーバに対して第三のクライアントが注文データを送信してしまう場合を示した図である。

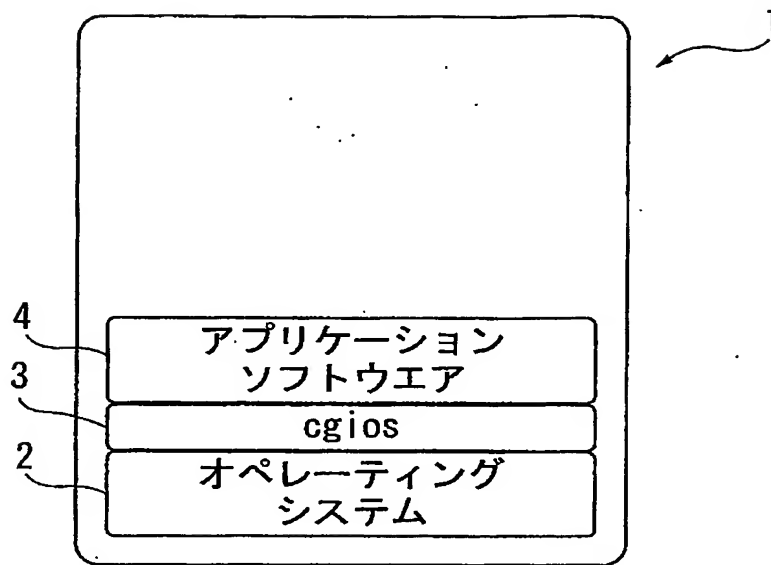
【符号の説明】

- 1、33、34 サーバ
- 2 オペレーティングシステム
- 3 cgi os (支援ソフトウェア、支援プログラム)
- 4 アプリケーションソフトウェア

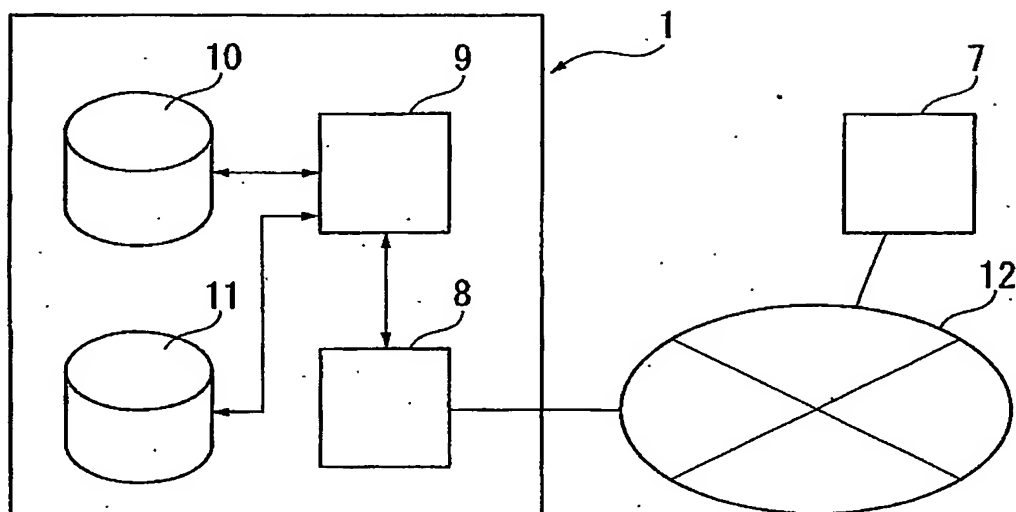
- 7、3 2 クライアント
- 8 送受信部（通信手段）
- 9 制御部（制御手段）
- 1 0 記憶部（記憶手段）
- 1 1 データベース
- 1 2 ネットワーク

【書類名】 図面

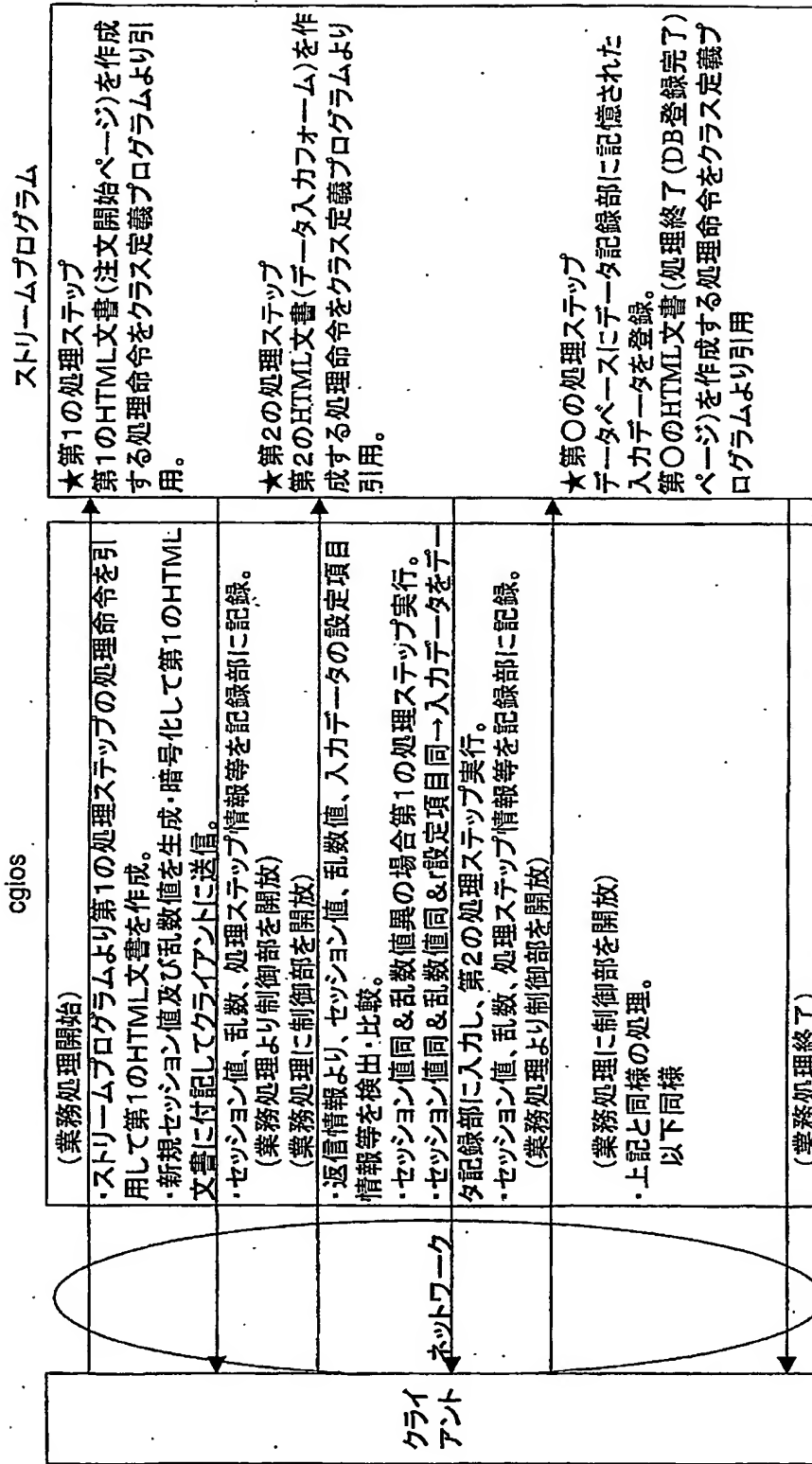
【図 1】



【図 2】



【図 3】



【図 4】

```
ustream input_order($o) { # 注文入カストリーム
  top:
    $o = new <APP>::OysterOrder($S);
    $o->default;

  first:
    page(input_order.page) {
      $V->{SLOT} = { slot => $o->input };
    }
    if ($V->{action} eq "next") { goto second; }
    goto first;

  second:
    if ($o->{ERROR} = $o->check) { goto first; }
    page(print_order_conf.page) {
      $V->{SLOT} = { slot => $o->print };
    }
    if ($V->{action} eq "back") { goto first; }
    elsif ($V->{action} eq "next") { goto third; }
    goto second;

  third:
    $o->{orid} = $o->nextorid;
    $$S{DA}->saveObject($o);

  fourth:
    page(print_order_done.page) {
      $V->{SLOT} = { slot => $o->print };
    }
    if ($V->{action} eq "next") { goto first; }
    goto fourth;
}
```

【図 5】

牡蛎料理の注文を表現するオブジェクト。メンバ変数の詳細は <MEMBER> である。

```
package OysterOrder;
```

```
# 牡蛎料理の注文を表現するオブジェクト
```

```
#BEGIN_DECLARE
```

```
$data = "orid int,
```

```
brand int,
```

```
recipe int,
```

```
trimming int,
```

```
comment text,
```

```
primary key(orid)";
```

```
# 注文 ID
```

```
# 牡蛎の銘柄コード
```

```
# 調理方法コード
```

```
# 調味料コード
```

```
# 調理方法についてのコメント
```

p1

```
$seqs = "seq_orid:orid:1,";
```

```
$defo = "brand => OysterType, recipe => RecipeType, trimming => TrimmingType";
```

```
#END_DECLARE
```

```
use strict;
```

```
sub new { # 牡蛎料理注文オブジェクトのコンストラクタ
```

```
my($type,$s,%hash) = @_;
```

```
my %h = ("S" => $s, "SELF" => "OysterOrder", %hash);
```

```
return bless \%h, "<APP>::OysterOrder";
```

```
}
```

p2

```
sub nextorid { # 次の ORID
```

```
my($V) = @_; my $S = $V->{S};
```

```
my($orid) = $$S{DA}->getrow("select nextval('seq_orid')");
```

```
$orid;
```

```
}
```

p3

```
sub default { # デフォルト値をセット
```

```
my($V) = @_; my $S = $V->{S};
```

```
$V->{brand} = 1; # 岩牡蛎
```

```
$V->{recipe} = 1; # 生食
```

```
$V->{trimming} = 2; # レモンのみ
```

```
$V->{comment} = "";
```

```
delete $V->{orid};
```

```
}
```

p4

【図 6】

```

sub input { # 注文入力フォーム
    my($V) = @_ ; my $S = $V->{S};
    print_webform( # 注文入力フォーム
    <table>
    <ifdef ERROR><span class=error><!ERROR></span></ifdef>
    <tr><td>牡蛎の銘柄</td><td>
        <input type=select name=brand value=$brand choice=[OysterType]></td></tr>
    <tr><td>牡蛎の調理方法</td><td>
        <input type=select name=recipe value=$recipe choice=[RecipeType]></td></tr>
    <tr><td>添える調味料</td><td>
        <input type=TrimmingType name=trimming value=$trimming></td></tr>
    <tr><td>コメント</td><td>
        <input type=text name=comment value=$comment size=60></td></tr>
    </table>');
}

```

p 5

```

sub check {
    my($V) = @_ ; my $S = $V->{S};
    my $count = 0;
    my $value = $V->{trimming};
    while ($value>0) { $count += $value & 1; $value >>= 1; }
    my $error = "";
    if ($count >= 3) {
        $error = "<i>同時に添付できる調味料は2種類までです。";
    }
    $error;
}

```

p 6

```

sub print { # 注文の HTML 形式での清書
    my($V) = @_ ; my $S = $V->{S};
    print_webtext( # 注文の HTML 形式での清書
    <table>
    <tr><td>牡蛎の銘柄</td><td><%brand></td></tr>
    <tr><td>牡蛎の調理方法</td><td><%recipe></td></tr>
    <tr><td>添える調味料</td><td><%trimming></td></tr>
    <tr><td>コメント</td><td><%comment></td></tr>
    </table>');
}

```

p 7

【図 7】

(a) < 注文を入力するページ:/main/input_order.page>

```
<h3>牡蛎の銘柄・調理方法を選択してください</h3>
<#slot>
<br>
<input type=submit action="next" label="次へ">
```

(b) < 注文を確認するページ:/main/print_order_conf.page>

```
<h3>御注文を確認ください。</h3>
<#slot>
<br>
<input type=submit action="next" label="注文実行">
<input type=submit action="back" label="戻る">
```

(c) < 注文の完了を通知するページ:/main/print_order_done.page>

```
<h3>以下の御注文を承りました</h3>
<#slot>
<br>
<input type=submit action="next" label="最初に戻る">
```


【图 8】

22

20

21

【図 9】

■ cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る ◀ ▶ 検索 🔍 お気に入り 📌 メディア 🖨️

アドレス (U) http://OOOO.net/oyster_r2/sp/kernel 移動 ➡

牡蛎の銘柄・調理方法を選択してください

牡蛎の銘柄 岩牡蛎 ▼

牡蛎の調理方法 生食 ▼

添える調味料 ☒ レモン ☒ カクテルソース ☐ ホースラディッシュ

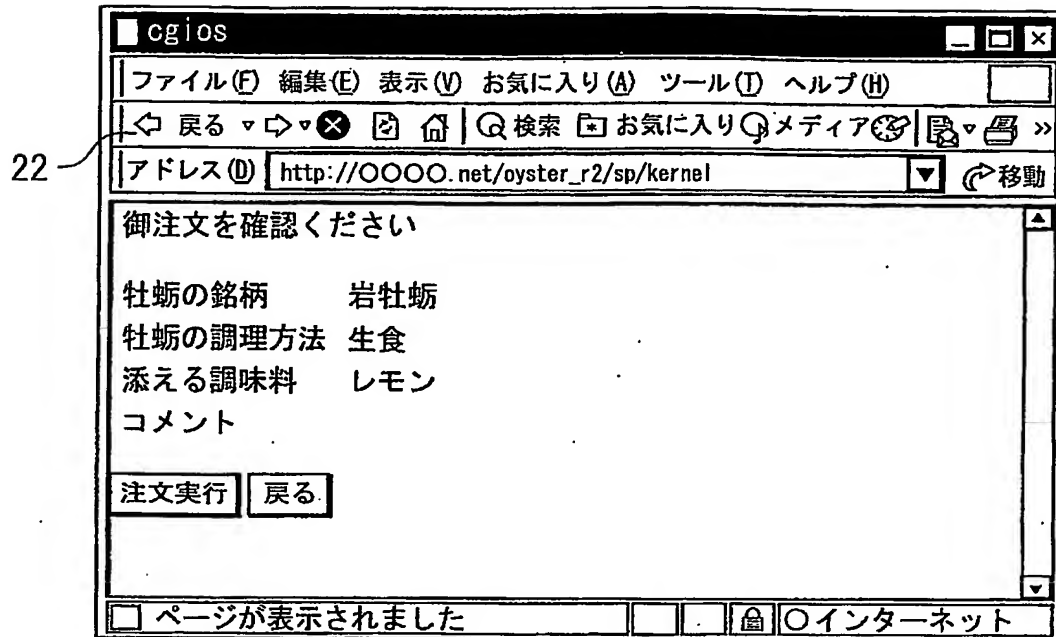
コメント

次へ

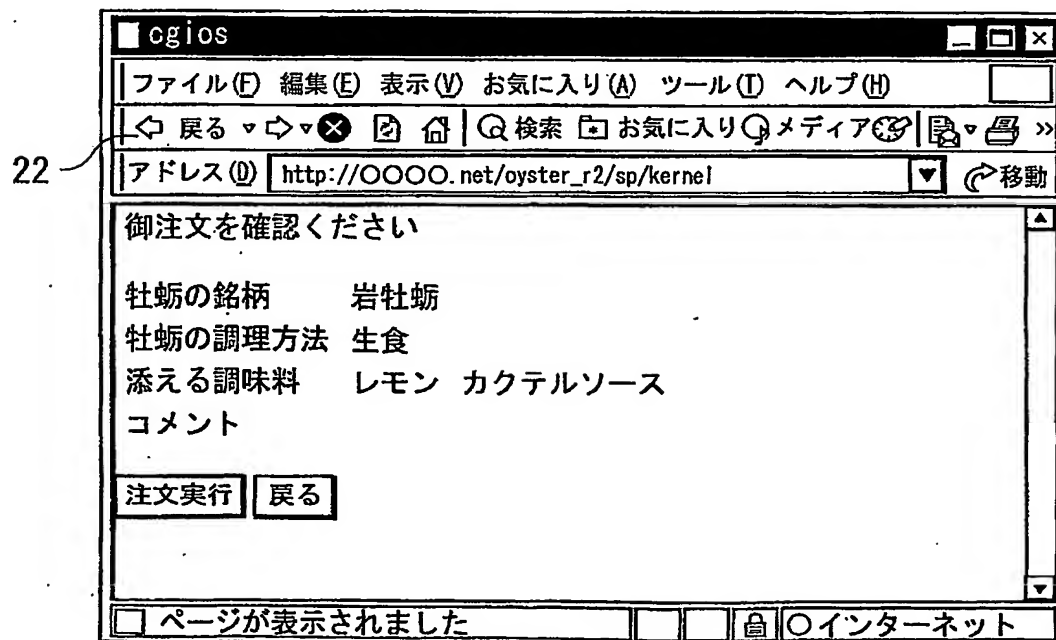
ページが表示されました

インターネット

【図10】



【図11】



【図 12】

22

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 停止 印刷 検索 入力 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

牡蛎の銘柄・調理方法を選択してください

・同時に添付できる調味料は2種類までです

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 ☒ レモン ☒ カクテルソース ☒ ホースラディッシュ

コメント

次へ

ページが表示されました

インターネット

【図 13】

22

20

21

cgios

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 停止 印刷 検索 入力 お気に入り メディア 移動

アドレス(D) http://OOOO.net/oyster_r2/sp/kernel

以下の御注文を承りました

牡蛎の銘柄 岩牡蛎

牡蛎の調理方法 生食

添える調味料 レモン

コメント

最初に戻る

ページが表示されました

インターネット

【図 14】

22

20

20

21

■ CgiOS

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 停止 印刷 検索 気に入りに追加 メディア 移動

アドレス (A) http://OOOO.net/oyster_r2/sp/kernel

牡蠣の銘柄・調理方法を選択してください

牡蠣の銘柄 岩牡蠣

牡蠣の調理方法 生食

添える調味料 ☒ レモン ☐ カクテルソース ☐ ホースラディッシュ

コメント

牡蠣の銘柄 岩牡蠣

牡蠣の調理方法 生食

添える調味料 ☒ レモン ☐ カクテルソース ☐ ホースラディッシュ

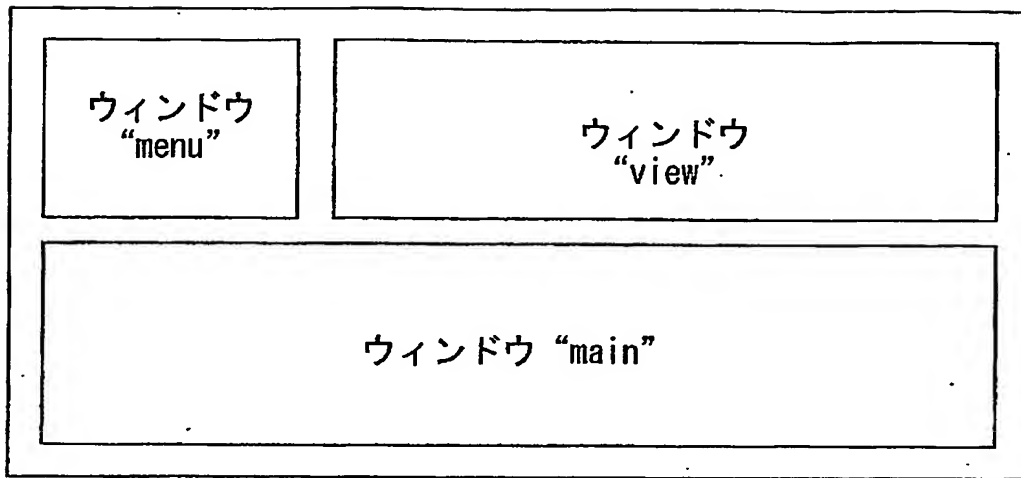
コメント

次へ

ページが表示されました

インターネット

【図 15】



フレーム "/main/top.page"

【図 16】

The screenshot shows a web browser window titled "cgios". The address bar displays "http://0000.net/oyster_r4/sp/kernel". The page content includes a "メインメニュー" (Main Menu) with links for "注文入力" (Order Input) and "注文一覧表示" (Order List Display). The "注文入力" section prompts the user to select oyster type and cooking method. The "注文一覧表示" section shows a message "該当する注文がありません" (No orders found) and buttons for downloading and uploading data in XLS format.

注文入力

牡蠣の銘柄・調理方法を選択してください

牡蠣の銘柄

牡蠣の調理方法

添える調味料 ☐ レモン ☐ カクテルソース ☐ ホースラディッシュ

コメント

次へ 閉じる

注文一覧表示

受注状況の一覧です

該当する注文がありません

xls形式でダウンロード 参照... xls形式でアップロード 閉じる

インターネット

【図 1 7】

(a) <起動ストリーム:/init.stream>

```
ustream init() { # 起動ストリーム
loop:
  p_fork("ウィンドウストリーム","/main/init.stream","this");
  p_wait();
  goto loop;
}
```

(b) <ウィンドウ起動ストリーム:/main/init.stream>

```
ustream init() { # ウィンドウストリーム
loop:
  p_setPage("top.page");
  p_fork("メニュー","menu.stream","menu");
  p_wait();
  goto loop;
}
```

(c) <メニューストリーム:/main/menu.stream>

```
ustream menu() { # メニュー表示ストリーム
first:
  page(menu.page) {
  }
  if ($V->{action} eq "input_order") {
    p_fork("注文入力","input_order.stream","^view");
  } elsif ($V->{action} eq "list") {
    p_fork("注文一覧表示","list.stream","^main");
  }
  goto first;
}
```

【図 1 8】

(a) <トップフレーム(新規追加):/main/top.page>

```

<table width=100% height=100% bgcolor=#ffffff border=0 cellspacing=2
cellpadding=0 style="table-layout:fixed">
  <tr>
    <td width=143 valign=top align=left class=wallpaper>
      <div style="overflow:auto"><#menu></div></td>
    <td valign=top class=wallpaper>
      <div style="overflow:auto"><#view></div></td>
  </tr>
  <tr>
    <td colspan=2 height=70% valign=top class=wallpaper>
      <div style="overflow:auto"><#main></div></td>
  </tr>
</table>

```

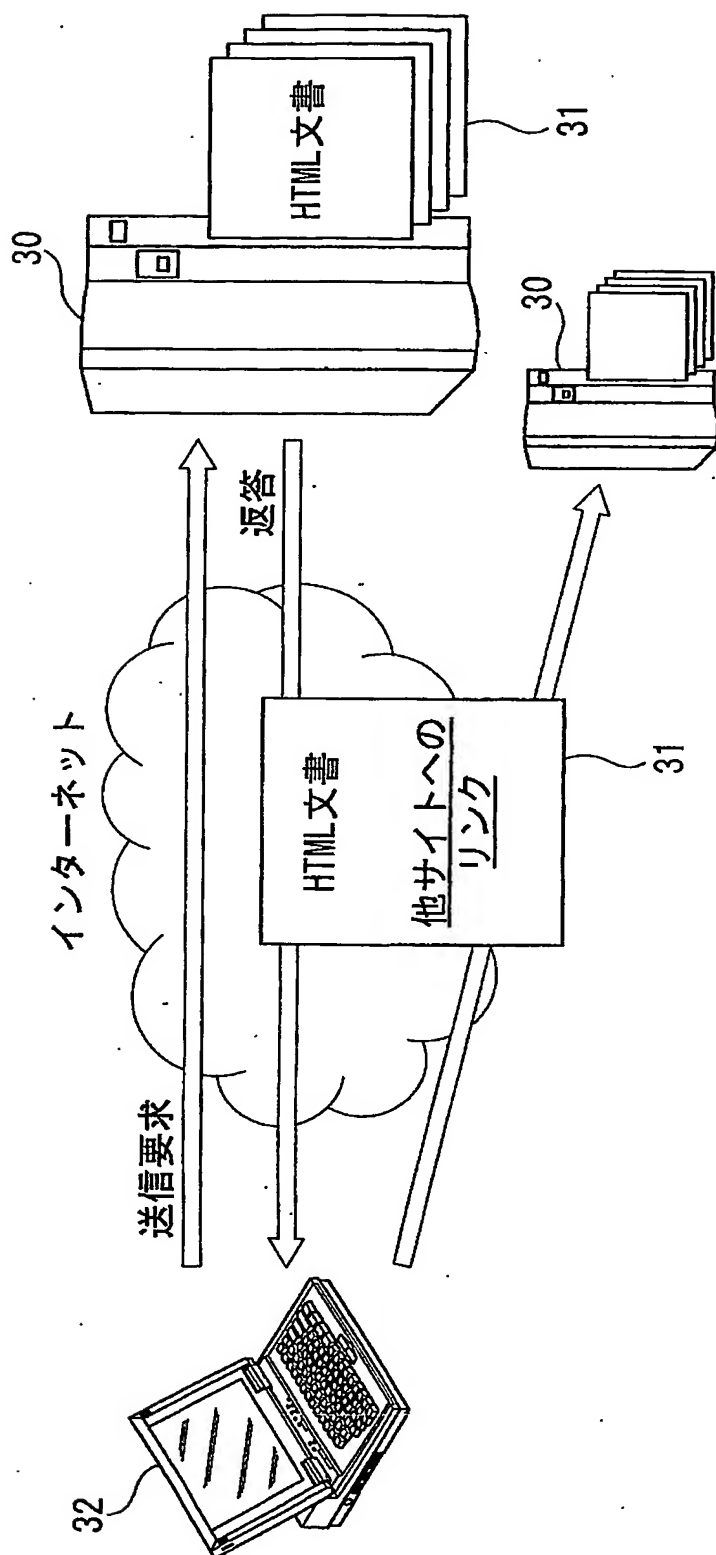
(b) <メニューページ(新規追加):/main/menu.page>

```

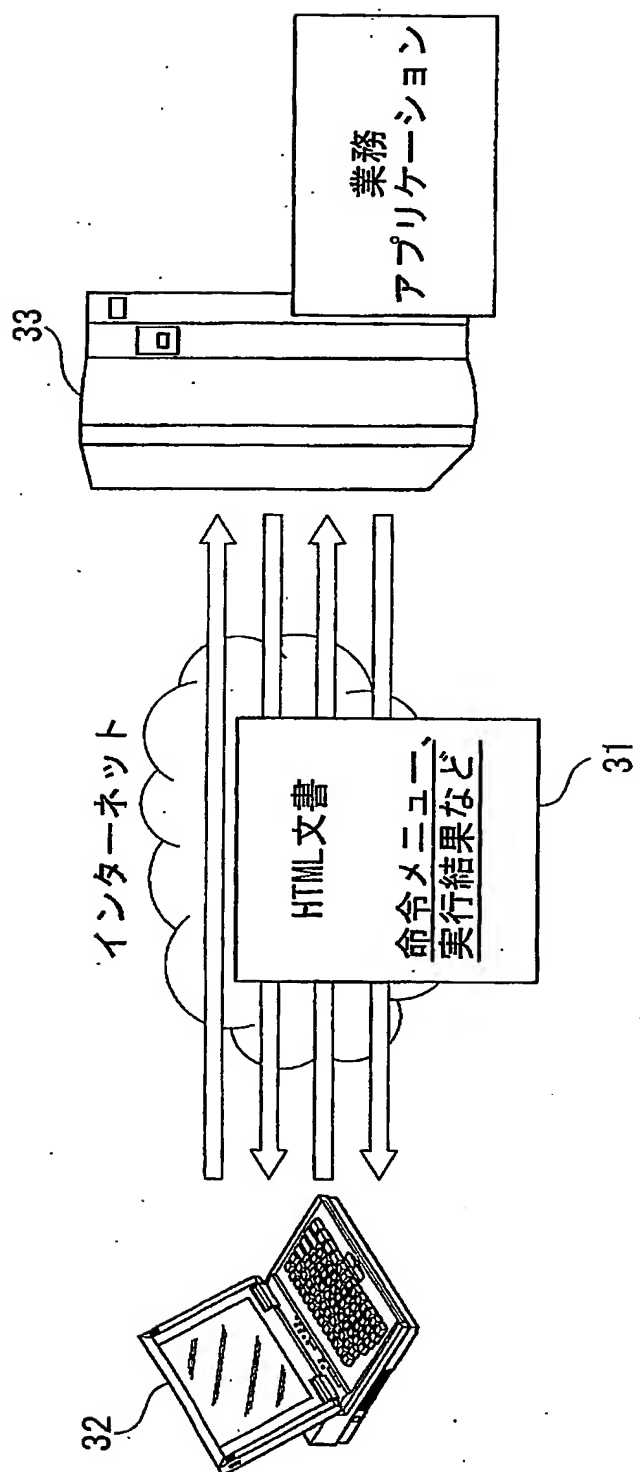
<h3>メインメニュー</h3>
<table><tr><td valign=top>
  <table class=inner bgcolor=#ffffff border-color=#000000 border=2
cellspacing=0 cellpadding=10>
    <tr><td valign=top>
      <li><a href="action:input_order">注文入力</a>
      <li><a href="action:list">注文一覧表示</a>
    </td></tr>
  </table>
</td></tr></table>

```

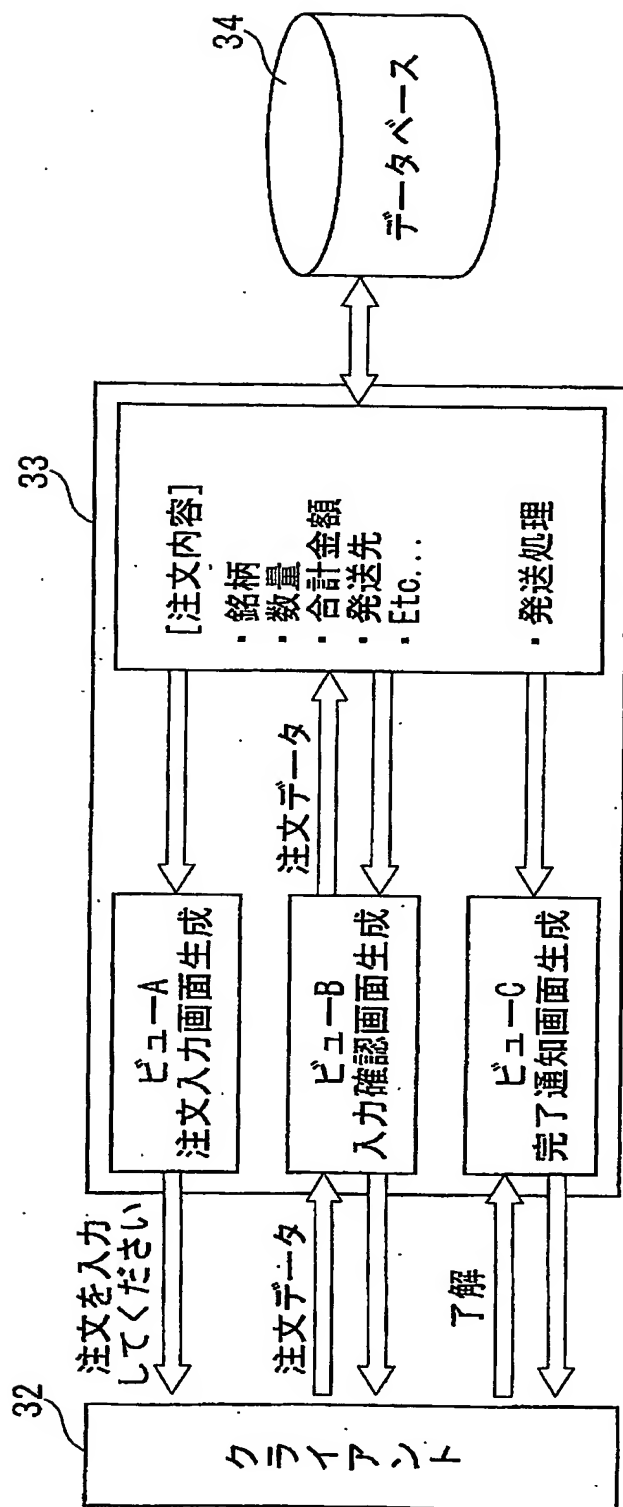
【図 19】



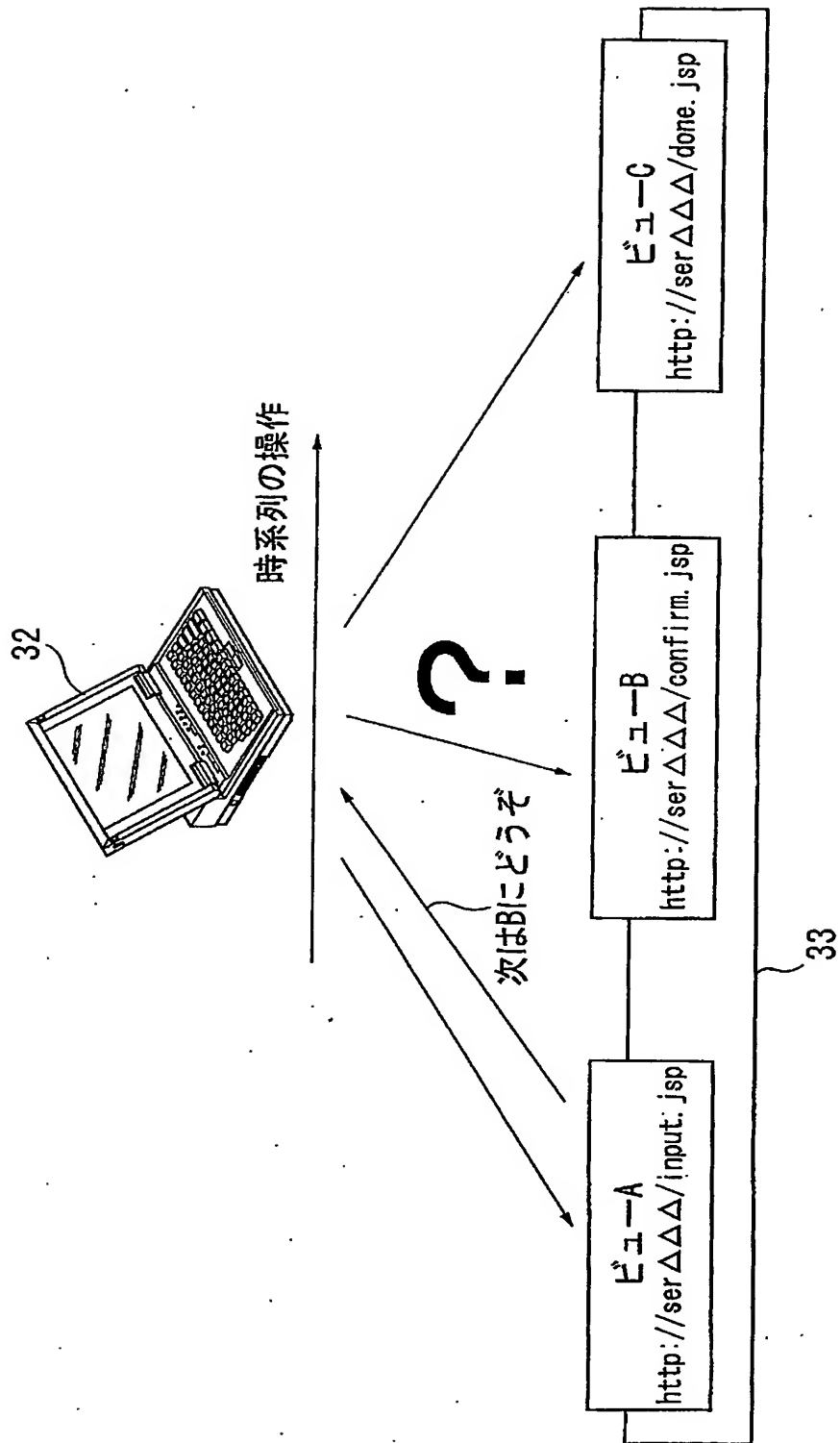
【図 20】



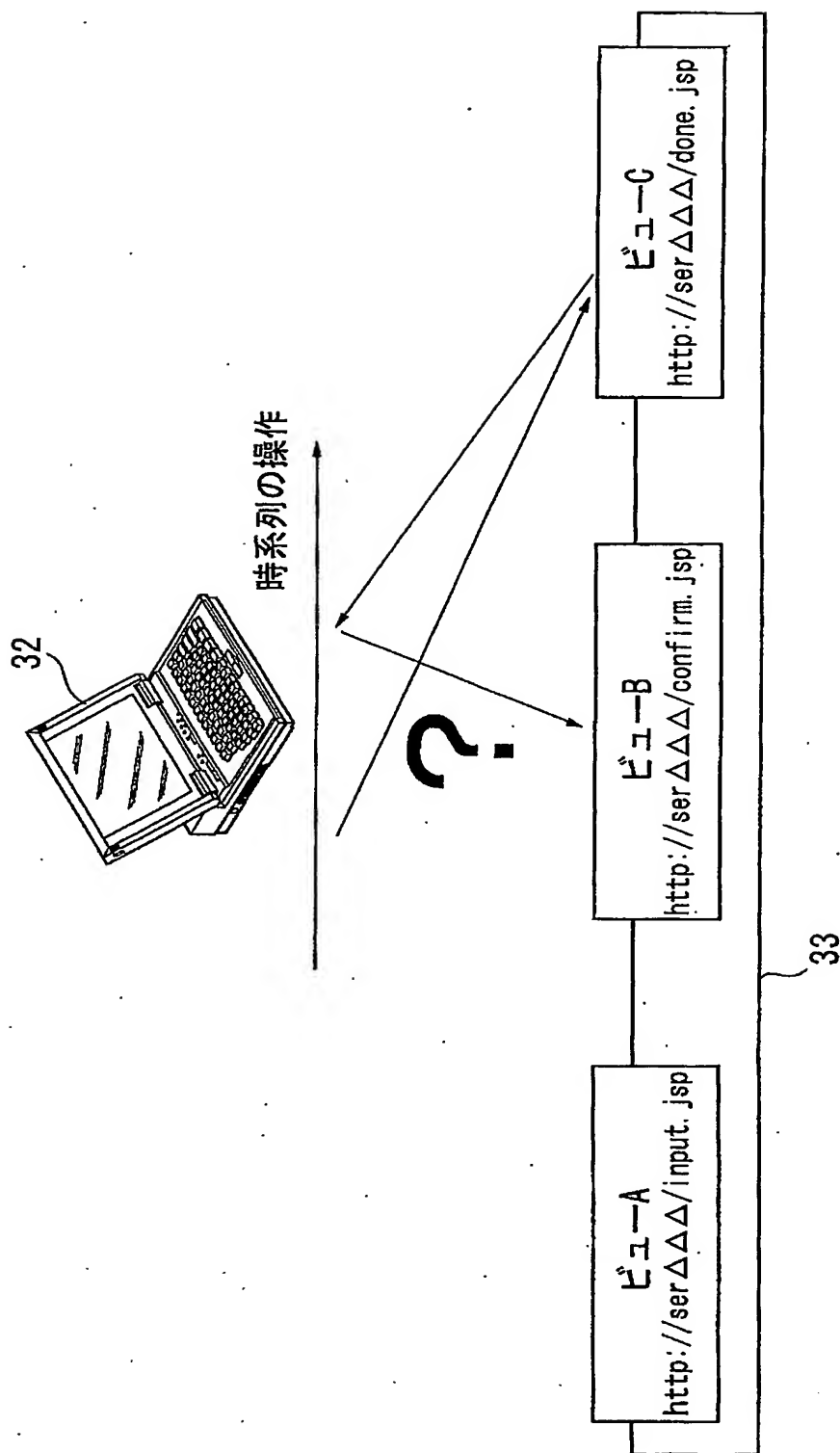
【図 21】



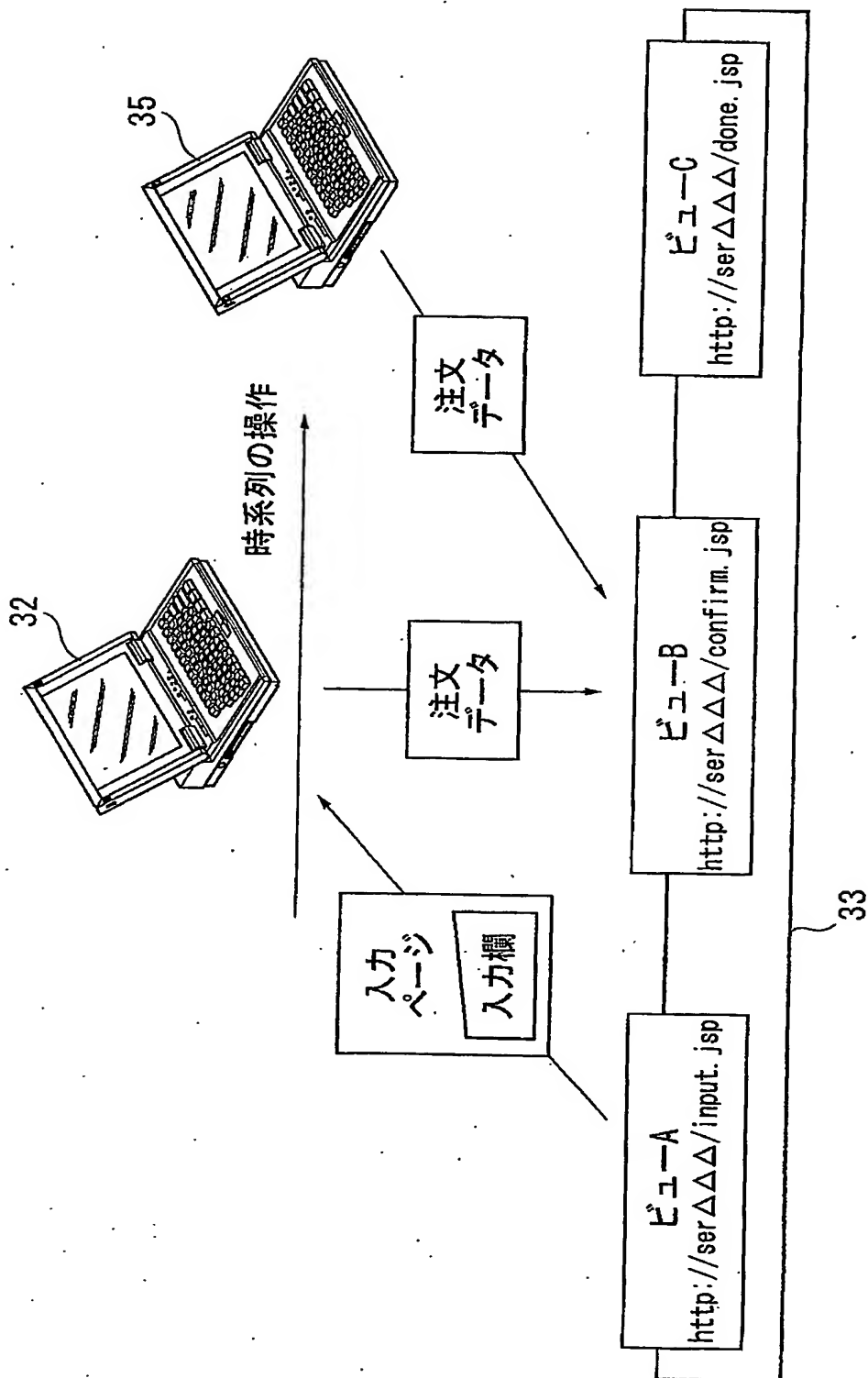
【図 22】



【図 23】



【図24】



【書類名】 要約書

【要約】

【課題】 一連の処理の処理過程を把握することができ、さらに、一連の処理に用いられるウェブ情報の内容を容易に変更することができるサーバを提供する。

【解決手段】 本発明に係るサーバ1は、クライアント7を特定するためのクライアント特定情報とウェブ情報を特定するためのウェブ特定情報とをウェブ情報に付加してクライアント7に送信するとともに、ウェブ情報に付加されたクライアント特定情報及びウェブ特定情報と、ウェブ情報が送信されたときの処理ステップに関する処理情報とを、記憶手段に記憶し、クライアント7より返信された返信情報にクライアント特定情報とウェブ特定情報とが含まれていないかを検出して処理情報より求められる処理ステップ又はその次の処理ステップを実行することを特徴とする。

【選択図】 図3

特願 2002-370953

出 願 人 履 歴 情 報

識別番号 [502248555]

1. 変更年月日	2002年 7月10日
[変更理由]	新規登録
住 所	東京都港区白金1-25-24-305
氏 名	澤田 研一